METHOD AND SYSTEM FOR RESPONSE TIME OPTIMIZATION OF DATA QUERY RANKINGS AND RETRIEVAL

Shamim A. Alpha

Field Of The Invention

[0001] The invention relates to the information retrieval and relevance ranking arts. It finds particular application to a method and system of optimizing the response time for ranking and retrieving documents from a search query. It will be appreciated that the present invention will find application in any domain where the final ranking of an object depends on one or more of its attributes.

Background Of The Invention

[0002] The World wide web has dramatically changed the requirements from information retrieval engines such as Oracle Text of the Oracle Corporation. Recent research shows that web users rarely look beyond the first two pages from a candidate hitlist with a total of twenty hits. Furthermore, users expect subsecond response time (regardless of the promised accuracy of the results). With these types of expectations, response time is of paramount importance. At the same time, since typical web users are not trained in information retrieval, it is imperative that search applications provide very forgiving syntax (or free text query) and deliver a reasonable hitlist.

[0003] A solution provided by Oracle Text is the ABOUT operator which accepts short free text queries and finds relevant documents using Oracle Text knowledge based linguistic retrieval system. The ABOUT operator internally uses ACCUMULATE (ACCUM) operator to rank queries with multiple nonstop words (stopwords are words like 'is', 'am', 'are', 'when' etc.). The response time and relevance ranking of the ABOUT query depends on the effectiveness of the ACCUM operator. One problem is slow response times for queries involving a few non-stopwords and unpredictable non-intuitive relevance rankings for queries involving more than one non-stopword. Both of these problems are attributable to the ACCUM operator scoring semantics.

[0004] In other prior information retrieval and ranking systems, even when a user is interested in only a few most relevant documents, the ranking system has to retrieve and evaluate an exact relevance score for every single candidate document identified by the search. In a query, the presence of a single non-restrictive term forces the system to evaluate an exact relevance score for an extremely high number of documents. This is required because the prior systems can not identify the most relevant documents until the scores for all the documents were computed. The reason for this problem is that there is no necessary relation between the final score range of a document and the number of children or total weight matched.

[0005] The present invention provides a new and useful method and system that optimizes the response time and relevance rankings for search queries that cures the above problems and others.

Summary Of The Invention

Ins a'

~~**[0006]** According to one embodiment of the present invention, a method of optimizing a response time for retrieving relevant documents from a set of candidate documents is provided. The candidate documents are identified in response to a search query where the search query includes one or more terms. A term weight is assigned to each of the terms. Documents are associated to a relevance score bin based on a total matched term weight where a document that matches a total term weight of M is associated to a more relevant score bin than a document that matches a total term weight less than M. A set of most relevant documents are then retrieved based on the association to the relevance score bins having a highest relevance score without retrieving other candidate documents.~~

~~**[0007]** According to another embodiment of the present invention, an information retrieval system is provided. The system includes logic for processing a search query that has one or more terms. A document retrieval logic identifies candidate documents that match the search query. A ranking logic assigns a term weight to each of the terms and associates each combination of matched term weights to a relevance score range. The ranking logic also groups the candidate documents based on the matched term weight where a document that matches a total term weight of M is~~

a' cont

~~associated to a more relevant score range than a document that matches a total term weight less than M. A retrieval logic then retrieves a set of relevant documents associated to the relevance score ranges having a greatest matched term weight without retrieving the candidate documents from other relevance score ranges.~~

[0008] One advantage of the present invention is that response time is improved for processing search queries.

[0009] Another advantage of the present invention is that the most relevant documents are identifiable without having to retrieve all candidate documents thus reducing the number of retrievals.

[0010] Still further advantages of the present invention will become apparent to those of ordinary skill in the art upon reading and understanding the following detailed description of the preferred embodiments.

Brief Description Of The Drawings

[0011] In the accompanying drawings which are incorporated in and constitute a part of the specification, embodiments of the invention are illustrated, which, together with a general description of the invention given above, and the detailed description given below, serve to example the principles of this invention.

[0012] Figure 1 is an exemplary overall system diagram of an information retrieval and ranking system in accordance with the present invention; and

[0013] Figure 2 is an exemplary methodology of retrieving and ranking documents in accordance with the present invention.

Detailed Description Of Illustrated Embodiment

[0014] The following includes definitions of exemplary terms used throughout the disclosure. Both singular and plural forms of all terms fall within each meaning:

[0015] "Document", as used herein, generally refers to an object being searched for. It includes but is not limited to one or more electronic documents, web pages, network addresses or links, database addresses or records, or any objects (text or non-text) that have one or more attributes that can be searched.

[0016] "Software", as used herein, includes but is not limited to one or more computer executable instructions, routines, algorithms, modules or programs including separate applications or from dynamically linked libraries for performing functions as described herein. Software may also be implemented in various forms such as a servlet, applet, stand-alone, plug-in or other type of application. Software can be maintained on various computer readable mediums as is known in the art.

[0017] "Logic", as used herein, includes but is not limited to hardware, software and/or combinations of both to perform a function.

[0018] "Network", as used herein, includes but is not limited to the internet, intranets, Wide Area Networks (WANs), Local Area Networks (LANs), and transducer links such as those using Modulator-Demodulators (modems).

[0019] "Internet", as used herein, includes a wide area data communications network, typically accessible by any user having appropriate software.

[0020] "Intranet", as used herein, includes a data communications network similar to an internet but typically having access restricted to a specific group of individuals, organizations, or computers.

[0021] Illustrated in **Figure 1** is an exemplary overall system diagram in accordance with the present invention. An information retrieval system **100** receives and processes search queries from a user system **105** that is trying to locate information on a network **110** or from a particular database **115**. The information retrieval system **100** includes, for example, a search engine that is a remotely accessible software program that lets a user perform keyword searches for information on the network. An exemplary retrieval system is Oracle Text. The retrieval system **100** may also include or use one or more pre-generated indexes **120** that associate keys to web pages, addresses, documents or other objects accessible through the network **110** or database **115** as is known in the art. In response to a search query, the retrieval system **100** identifies a candidate set of documents that match or possibly match the criteria of the search query.

[0022] The information retrieval system **100** is embodied as software executable by a computer system. The computer system (not shown) generally may take many forms, from a configuration including a variety of processing units, networked together to function as a integral entity, to a single computer, e.g., a personal computer, operational in a stand-alone environment. The present invention can be embodied in any of these computer system configurations. As known in the art, computer systems may include a variety of components and devices such as a processor, memory, data storage, data communications buses, and a network communications device.

[0023] A search query typically includes one or more keywords, phrases, or attributes (text or non-text) that represent subject matter or content that a user wishes to locate. A keyword, phrase or attribute will be referred to as a "term." When a search query is received, a parser **125** parses the query to identify the terms. If desired, certain terms can be eliminated such as pronouns or prepositions. The present invention is particularly suited for queries that are in a free form that allows a user to simply list one or more terms with little or no required syntax. Typical free text queries match documents that contain any one of the query terms. In other words, the search applies an "OR" operation between the query terms. This may result in a very large set of candidate documents to be retrieved, thus, resulting in a slower response time and retrieving documents that have questionable relevance ranking scores.

[0024] With further reference to **Figure 1**, document retrieval logic **130** identifies documents from the index database that match the search query and determines intersections of documents matching multiple terms. A ranking logic **135** ranks each document based on its relevance to the search query and the documents are displayed to the user as a hitlist ordered by the most relevant documents. Ranking includes computing, for each document, a term relevance score based on the document's relevance to each matched term and a term weight for each term. The term relevance score can be computed in many ways to reflect the relevance of a document's content to a term. It may be based on, for example, a number of occurrences of the term in the document, a link analysis of pages that reference the document, font analysis of text within the document, etc.

[0025] The term weight for a term is based on an occurrence frequency of the term. In particular, a term that is uncommon (e.g. occurs infrequently) is given a greater term weight than a term that is common (e.g. occurs frequently). Thus, a document that matches an uncommon query term should be more relevant than if the document only matches a common query term. A known technique for determining term weight in this manner is inverse frequency scoring. This technique adjusts the relevance of a document by increasing its relevance if the document matches an uncommon term.

Ins a2)

~~**[0026]** A final relevance score for a document is a function of the term relevance score and the term weight for each term the document matches. However, before the final relevance score is computed, the ranking logic **135** establishes a relationship between the total term weight matched by a document and a range of final relevance scores that can be assigned to the document. Thus, the term weight is a primary factor for determining the~~

a3 con't

~~final relevance score of a document. In this manner, a document that matches a term weight of M from the query will always have a greater relevance score than a document that matches a term weight less than M. This is described in greater detail below. As a result of this relationship, the most relevant documents can be identified without having to compute the relevance scores of the entire set of candidate documents. Thus, a small set of most relevant documents can be retrieved without having to retrieve the entire set of candidate documents thereby increasing response time.~~

[0027] Illustrated in **Figure 2** is an exemplary computer-implemented methodology of processing a search query, ranking and retrieving candidate documents for relevance in accordance with the present invention. The blocks shown represent functions, actions or events performed therein. It will be appreciated that computer software applications involve dynamic and flexible processes and logic such that the illustrated blocks can be performed in other sequences different than the one shown. It will also be appreciated by one of ordinary skill in the art that the software of the present invention may be implemented using various programming approaches such as procedural, object oriented or artificial intelligence techniques.

[0028] With reference to **Figure 2**, the process will be described using an exemplary search query of "oracle text adoption in Japan." The system receives the search query (block **200**). For example, if using Oracle Text, the search query may be in the form "ABOUT(phrase)" where "phrase" can be a single word or a phrase, or a string of words in a free text format. Using the example query, it reads "ABOUT(oracle text adoption in Japan)." The ABOUT operator retrieves documents that contain themes or words that match the terms of the query. Typically, a database will have a theme component in its index to obtain better results using the ABOUT operator but it is not required. The search query is then parsed to identify terms, for example, term1 is "oracle text", term2 is "adoption" and term3 is "Japan" (block **205**).

[0029] Once the terms are identified, the system identifies documents that match one or more of the terms or their themes (block **210**) and builds a hitlist for each term. The hitlist can be a table, index, tree or the like that identifies which documents matched each term. Suppose that the selectivity of the terms are as follows in Table (1):

Table (1):

Term	Document Hits
Oracle Text:	1,000
Adoption:	50,000
Japan:	50,000,000

[0030] Each term is assigned a term weight based on its occurrence frequency (block **215**). Inverse frequency scoring is one algorithm that can be used. Terms that occur infrequently are assigned a greater term weight than terms that are occur frequently. For this example, the term weights are assigned as 4 for "oracle text," 2 for "adoption," and 1 for "Japan."

[0031] By applying the term weights, it means that matching an uncommon word from the query is more important than matching a common word from the query. Since in the example above, Japan is a very common word compared to "oracle text", it is assumed that the user is more interested in finding a document that talks more about "oracle text" than a document that talks more about Japan.

[0032] Intersections are then determined between the documents to find which documents match more than one query term (Block **220**). Let us assume that the selectivity of the intersections of the terms are as follows in Table (2):

Table (2):

Term Intersections	Document Hits
oracle text & adoption:	30
oracle text & Japan:	100
adoption & Japan:	1,000
oracle text & adoption & Japan:	5

[0033] A document receives a final relevance score as a function of each matched term relevance score and its term weight. The initial determination, however, is based on a document's matched term weights. A total score range is predetermined, for example, 0-100 where a score of 100 means a document is most relevant. Of course, other score ranges can be used. The total score range is divided into score bins as a function of the possible term weights that can be matched. An exemplary function is shown in Equation (1) below. For the present example, a simplified function is used as follows: the total possible term weights a document can have is seven (based on combinations of term weights 4, 2 and 1), so the total score range is divided into seven (7) score bins, one for each matched term weight value. Thus, the total score range of 100 is divided by seven (7). This defines seven (7) score bins each having a score range of about fourteen (14) values.

[0034] With further reference to **Figure 2**, the documents are grouped together based on the sum of matched term weights (Block **225**). Each document group is then associated to one of the predefined score ranges based on their matched term weights (Block **230**). Using the above example of matched documents and matched terms, the following Table (3) is an exemplary classification of document groups associated to a particular score range:

Table (3):

doc range	# of docs	matched terms	matched weight	score range
1-5	5	all	7	87 - 100
6-30	25	OT, adoption	6	71 - 86
31-125	95	OT, Japan	5	57 - 70
126-1000	875	OT	4	43 - 56
1001-1995	995	adoption, Japan	3	29 - 42
1996-50970	48975	adoption	2	15 - 28
50971-5050875	49999905	Japan	1	1 - 14

[0035] As shown in Table (3), there are five (5) documents that matched all the terms. Thus, their matched term weight is 4+2+1=7 and these documents are assigned to the top score range of 87-100. For the documents that matched only "oracle text" (OT) and "Japan," their matched term weight is 4+1=5 and these documents are assigned to the fifth score bin having a range of 57-70.

[0036] In effect, documents are pre-classified into score ranges before actual relevance scores are computed for each document. By associating a document to a score range based on its matched term weight, documents matching weight M+1 are guaranteed to have a total relevance score more than documents matching weight M. In the above table, the group of five documents matching weight of 7 is guaranteed to contain the top five documents of the collection of candidate documents. The next 25 documents matching weight 6 are guaranteed to be the next 25 most relevant documents and so on.

[0037] With this pre-scoring association, the system does not have to retrieve all candidate documents or compute a total relevance score for every candidate document before it can determine which documents are most relevant. Rather, the system can process groups of documents based on decreasing matched weight. In the example above, if the system needs to find the top thirty (30) documents and return them to the user, it needs to evaluate the total relevance score for only thirty (30) documents. In this case, the top thirty documents are already identified in the top two scoring bins matching term weights of 6 and 7. This results in less processing time, less disk accesses and ultimately faster response time because the system does not need to retrieve and evaluate the total relevance score for all 5,050,875 candidate documents before documents having the top thirty relevance scores are found. Additionally, more accurate relevance scores are obtained.

[0038] With reference again to **Figure 2**, after the documents are grouped into score ranges, a predetermined number N of most relevant documents are returned to the user as a resultant hitlist for the search query. From the score ranges based on decreasing term weight, a total relevance score is computed for each the top N documents to determine the relevance order between them (Block **235**). An exemplary total relevance score is computed in Equation (1) as follows:

$$\text{score} = \min(s', 100) \qquad \textbf{(1)}$$
$$\text{given } s' = \text{round}(a' + b')$$
Where
$$a' = (\,(\text{sum of matched weight}) - (\text{gcd of weights}))\ {*}100/\text{sum }(W_i)$$
$$b' = \max(1, (\text{sum}(W_i{*}S_i)/(\text{sum of matched weights}))\ {*}\ (\text{gcd of weights})/\text{sum }(W_i)$$

Ins A3

~~**[0039]** As described above, using a total scoring range of 0-100, it is split into equal sized bins (or substantially equal size) as a function of the sum of term weights. For example, it can be based on (sum of term weights)/(gcd of weights) where "gcd" is the greatest common denominator. Using the gcd allows the system to find an optimal size of score bin. Based on the total term weight a document matches, the relevance score for the document is confined within a specific bin. Based on the scores for individual terms, the relevance score of the document is placed at a specific position within the bin. The value of a' determines which score range (bin) a document is assigned to and the value of b' determines the position of the document within that bin. Of course, other relationships can be used and will be appreciated by those of ordinary skill in the art.~~

[0040] For example, assume a document matches two terms, "adoption" and "Japan" and that the term relevance score for "adoption" = 50 and for "Japan" = 60. The term weight is 2 for "adoption" and 1 for "Japan." Thus, out of a possible total term weight of 7, the document has a total matched term weight of 3 putting the document in the 29-42 score range (using the above Table (3)). The final relevance score for a document can be found as:

$$\text{(Score Range) } a' = \frac{(3-1)*100}{7} = 28.57 \qquad \textbf{(2)}$$

$$\text{(Location within range) } b' = \frac{(2*50+1*60)/3}{7} = 7.61 \qquad \textbf{(3)}$$

$$\text{Final Score} = \text{Round } (a' + b') = 36 \qquad \textbf{(4)}$$

[0041] Since the expected range of the score is determined based on the weight matched by the documents, a list of candidate documents can be isolated for top N hits by finding M (M>N) documents with highest matched weights. The final relevance scores are computed for the top N documents and the results are displayed to the user (Block **240**), for example, by displaying links, addresses and/or other identifying information for each document. If only 5 most relevant documents are desired, the system only needs to retrieve the 5 documents associated to the highest score range which in this case are the 5 documents that match all three terms.

[0042] In another example using the Oracle Text system, the intital query ABOUT(oracle text adoption in Japan) is rewritten by the system to use an ACCUM operator as follows:

ABOUT (oracle text)*4 ACCUM about (adoption)*2 ACCUM about (japan) *1.

[0043] The "ACCUM" (ACCUMULATE) operator is an instruction to search for documents that contain at least one occurrence of any query term. Both the ABOUT and ACCUM operators are well known in Oracle systems. The ACCUM operator returns a relevance score for a document where the documents that contain the most occurrences for the highest number of query terms are assigned the highest score. The "*number" indicates a term weight, e.g. "*4". Incorporating the present invention, the score returned by the ACCUM operator is set to the score shown in Equation (1) above.

Ins a4>

~~[0044] It will be appreciated by those of ordinary skill in the art that other scoring formulas can be used. The principle association is to associate a document's matched term weight to a particular score range. It will also be appreciated that the sequence of processing described above is only for exemplary purposes and that other sequences can implemented. It will further be appreciated that the present invention is not limited to the text/information retrieval domain. Rather, it applies to any domain where the final ranking of an object depends on one or more of its attributes whether equally or unequally weighted). If some of the attributes can receive a score of zero for some of the objects, the present invention will work as well. In this manner, the information retrieval system **100** would be generically an object retrieval system.~~

[0045] With the present invention, a relationship is established between the total weight matched by a document and the range of the final score the document receives. Thus, the relevance for a document and its relevance in relation to all candidate documents are identifiable independently from the relevance of the other candidate documents. This provides for less computations, less document retrievals and, thus, faster response times for identifying and retrieving a set of most relevant documents from a large candidate set.

[0046] While the present invention has been illustrated by the description of embodiments thereof, and while the embodiments have been described in considerable detail, it is not the intention of the applicants to restrict or in any way limit the scope of the appended claims to such detail. Additional advantages and modifications will readily appear to those skilled in the art. For example, relevance score ranges and rankings can be reversed where most relevant documents are assigned lower scores rather than higher scores. Term weights can be determined by other algorithms instead of inverse frequency scoring. Searching can be performed with or without indexes. Therefore, the invention, in its broader aspects, is not limited to the specific details, the representative apparatus, and illustrative examples shown and described. Accordingly, departures may be made from such details without departing from the spirit or scope of the applicant's general inventive concept.